Algebraic synchronization criterion and computing reset words

Mikhail Berlinkov[1] and Marek Szykuła[2]

[1] Institute of Mathematics and Computer Science, Ural Federal University, Russia
[2] Institute of Computer Science, University of Wrocław, Poland

Abstract. We refine a uniform algebraic approach for deriving upper bounds on reset thresholds of synchronizing automata. We express the condition that an automaton is synchronizing in terms of linear algebra, and obtain upper bounds for the reset thresholds of automata with a short word of a small rank. The results are applied to make several improvements in the area.

We improve the best general upper bound for reset thresholds of finite prefix codes (Huffman codes): we show that an n-state synchronizing decoder has a reset word of length at most $O(n\log^3 n)$. In addition to that, we prove that the expected reset threshold of a uniformly random synchronizing binary n-state decoder is at most $O(n\log n)$. We also show that for any non-unary alphabet there exist decoders whose reset threshold is in $\Theta(n)$.

We prove the Černý conjecture for n-state automata with a letter of rank at most $\sqrt[3]{6n-6}$. In another corollary, based on the recent results of Nicaud, we show that the probability that the Černý conjecture does not hold for a random synchronizing binary automaton is exponentially small in terms of the number of states, and also that the expected value of the reset threshold of an n-state random synchronizing binary automaton is at most $n^{3/2+o(1)}$.

Moreover, reset words of lengths within all of our bounds are computable in polynomial time. We present suitable algorithms for this task for various classes of automata, such as (quasi-)one-cluster and (quasi-)Eulerian automata, for which our results can be applied.

Keywords: Černý conjecture, Eulerian automaton, Huffman code, one-cluster automaton, prefix code, random automaton, reset word, reset threshold, synchronizing automaton

1 Introduction

We deal with *deterministic finite automata* (*DFA*) $\mathscr{A} = (Q, \Sigma, \delta)$, where Q is a non-empty *set of states*, Σ is a non-empty *alphabet*, and $\delta\colon Q\times\Sigma \mapsto Q$ is the complete *transition function*. We extend δ to $Q \times \Sigma^*$ and $2^Q \times \Sigma^*$ as usual, and for the image (resp. preimage) of a set S under a word w we write shortly $S.w$ (resp. $S.w^{-1}$). We denote $\Sigma^{\leq c} = \{w \in \Sigma^*\colon |w| \leq c\}$, the set of all words over Σ of length at most c. The empty word is denoted by ε. Throughout the paper, by n we denote the cardinality $|Q|$, and by k we denote $|\Sigma|$.

A word w *compresses* a subset $S \subseteq Q$ if $|S.w| < |S|$. Then we say that S is *compressible*. The *rank* of a word w is $|Q.w|$. A *reset word* or a *synchronizing word* is a word $w \in \Sigma^*$ of rank 1, that is, w takes the automaton to a particular state no matter of the current state. An automaton is called *synchronizing* if it possesses a reset word. An example of a synchronizing automaton from the Černý series [14] is presented in Fig. 1(left). One can verify that its shortest reset word is ba^3ba^3b. The length of the shortest reset word is called the *reset threshold* and is denoted by $\mathrm{rt}(\mathscr{A})$.

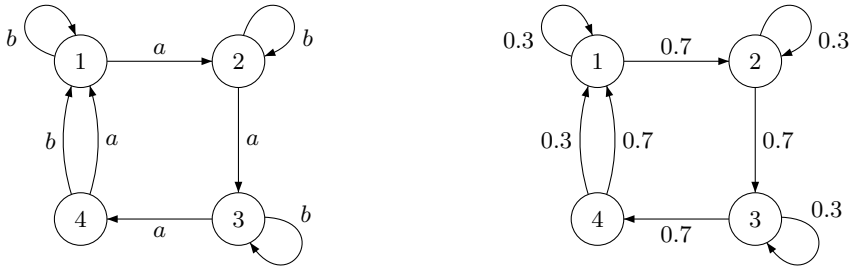

Fig. 1. The automaton $\mathscr{C}_4$ and the associated Markov chain for $P(a) = 0.7, P(b) = 0.3$.

Synchronizing automata serve as transparent and natural models of various systems in many applications in various fields, such as coding theory, DNA-computing, robotics, testing of reactive systems, and theory of information sources. They also reveal interesting connections with symbolic dynamics, language theory, and many other parts of mathematics. For a detailed introduction to the theory of synchronizing automata we refer the reader to the surveys [22,31], and for a review of relations with coding theory to [20].

In various applications, reset words allow to reestablish the control under the system modeled by an automaton. The reset threshold of an automaton serves as a natural measure of synchronization. Naturally, the shorter reset word the better, so from both theoretical and practical points of view it is important to compute the reset threshold, and shortest or short enough reset words.

In 1964 Černý [14] constructed for each $n > 1$ a synchronizing automaton $\mathscr{C}_n$ with n states and 2 input letters whose reset threshold is $(n-1)^2$. The automaton $\mathscr{C}_4$ is shown in Fig. 1(left). Soon after that he conjectured that those automata represent the worst possible case, thus formulating the following hypothesis:

Conjecture (Černý). *Each synchronizing automaton $\mathscr{A}$ with n states has a reset word of length at most $(n-1)^2$, i.e.* $\mathrm{rt}(\mathscr{A}) \leq (n-1)^2$.

By now, this simply looking conjecture is arguably the most longstanding open problem in the combinatorial theory of finite automata. Moreover, the best upper bound known so far for the reset threshold of a synchronizing n-state automaton is equal to $\frac{n^3-n}{6} - 1$ (for $n \geq 4$) and so is cubic in n (see Pin [28]). Thus it is of certain importance to prove better specific upper bounds for various important classes of synchronizing automata.

In this paper, we improve several results concerning reset thresholds. First, we express the condition that an automaton is synchronizing in terms of linear algebra, and derive upper bounds for automata with a word of a small rank (Section 2). Then, we apply the results to improve upper bounds in several cases.

We apply the results to improve upper bounds in several cases. In Section 4 we show that the Černý conjecture holds for automata with a letter of rank $\sqrt[3]{6n-6}$, which improves the previous logarithmic bound [27]. Also, basing on the recent results of Nicaud [25], we show that the Černý conjecture holds for a random synchronizing binary automaton with probability exponentially (in n) close to 1, and that the expected reset threshold is at most $n^{3/2+o(1)}$.

The next important application of our results is an upper bound for the length of the shortest reset words of decoders of finite prefix codes (Huffman codes), which are one of the most popular methods of data compression. One of the problems with compressed data is the reliability in case of

presence of errors in the compressed text. Eventually, a single error may possibly destroy the whole encoded string. One of the solutions to this problem (for Huffman codes) is a use of codes whose decoders can be synchronized by a reset word, regardless of the possible errors. Then, by inserting reset words into compressed data, we make the data error-resistant to some extent.

The reset thresholds of binary Huffman codes was first studied by Biskup and Plandowski [10,11], who proved a general upper bound of order $O(n^2 \log n)$. They also showed that a word of this length can be computed in polynomial time. The bound was later improved to $O(n^2)$ for a wider class of *one-cluster* automata [3]. In Section 5 we prove an upper bound of order $O(n \log^3 n)$. Next, we consider random decoders, and show that the expected reset threshold of a uniformly random synchronizing n-state binary decoder is at most $O(n \log n)$. We also show a series of decoders with linear reset thresholds over any alphabet of size at least 3. Such series were known only for a binary alphabet [11].

Unlike the general case, the Černý conjecture has been approved for various classes of automata such as circular [15], Eulerian [21], and one-cluster automata with prime length cycle [30]. Later, specific quadratic upper bounds for some generalizations of these classes were obtained in [3,4,7]. However, no efficient algorithm for finding reset words with lengths within the specified bounds has been presented for these classes. Moreover, there is no hope to get a polynomial algorithm for finding the shortest reset words in the general case, since this problem has been shown to be $\mathrm{FP}^{\mathrm{NP[log]}}$-hard [26]. Also, unless P = NP, there is no polynomial algorithm for computing the reset threshold for a given automaton within the approximation ratio $n^{1-\varepsilon}$ for any $\varepsilon > 0$, even in the case of a binary alphabet [18] (cf. also [8,9,19]).

In Section 3 we present polynomial algorithms for finding reset words of length guaranteed to be within the proven bounds. In particular, our algorithms can be applied to the classes of decoders of finite prefix codes, and also to generalized classes of quasi-Eulerian and quasi-one-cluster automata. Since from our results it is possible to derive the bounds from [3,4,7,12,21,29,30]), our algorithms apply to them as well.

A preliminary version of some of these results previously appeared in [5].

2 Algebraic synchronization criterion

In this section we refine some results from [7], formulate the algebraic synchronization criterion, and derive upper bounds for reset thresholds of automata with a word of a small rank. For this purpose, we associate a natural linear structure with an automaton $\mathscr{A}$. By $\mathbb{R}^n$ we denote the real n-dimensional linear space of row vectors. Without loss of generality, we assume that $Q = \{1, 2, \ldots, n\}$ and then assign to each subset $K \subseteq Q$ its *characteristic vector* $[K] \in \mathbb{R}^n$, whose i-th entry is 1 if $i \in K$, and 0, otherwise. For $q \in Q$ we write $[q]$ instead of $[\{q\}]$ to simplify the notation. By $\langle S \rangle$ we denote the linear span of $S \subseteq \mathbb{R}^n$. The $n \times n$ identity matrix is denoted by I_n.

Each word $w \in \Sigma^*$ corresponds to a linear transformation of $\mathbb{R}^n$. By $[w]$ we denote the matrix of this transformation in the standard basis $[1], \ldots, [n]$ of $\mathbb{R}^n$. For instance, if $\mathscr{A} = \mathscr{C}_4$ from Figure 1 (left), then

$$[a] = \begin{pmatrix} 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \\ 1&0&0&0 \end{pmatrix}, \ [b] = \begin{pmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \\ 1&0&0&0 \end{pmatrix}, \ [ba] = \begin{pmatrix} 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \\ 0&1&0&0 \end{pmatrix}.$$

Clearly, the matrix $[w]$ has exactly one non-zero entry in each row. In particular, $[w]$ is *row stochastic*, that is, the sum of entries in each row is equal to 1. In virtue of row-vector notation (apart from [7]), we get that $[uv] = [u][v]$ for every two words $u, v \in \Sigma^*$. By $[w]^T$ we denote the transpose of the

matrix $[w]$. One easily verifies that $[S.w^{-1}] = [S][w]^T$. Let us also notice that within this definition the (adjacency) matrix of the underlying digraph of $\mathscr{A}$ is equal to $\sum_{a\in\Sigma}[a]$.

Recall that a word w is a reset word if $q.w^{-1} = Q$, for some state $q \in Q$. Thus, in the language of linear algebra, we can rewrite this fact as $[q][w]^T = [Q]$. For two vectors $g_1, g_2 \in \mathbb{R}^n$, we denote their usual inner (scalar) product by (g_1, g_2). We say that a vector (matrix) is *positive* (*non-negative*) if it contains only positive (non-negative) entries. Let $p \in \mathbb{R}^n_+$ be a positive row stochastic vector. Then $([Q], p) = 1$, and a word w is a reset word if and only if

$$([q.w^{-1}], p) = ([q][w]^T, p) = ([q], p[w]) = 1.$$

Now we need to recall a few properties of Markov chains. A *Markov chain* of an automaton $\mathscr{A}$ is the random walk process of an agent on the underlying digraph of $\mathscr{A}$ where each time an edge labeled by a_i is chosen according to a given probability distribution $P\colon \Sigma \mapsto R$. The matrix $S(\mathscr{A}, P) = \sum_{i=1}^{k} P(a_i)[a_i]$ is called the *transition matrix* of this Markov chain. An example of a Markov chain associated with the automaton $\mathscr{A} = \mathscr{C}_4$ is presented in Figure 1 (right) for $P(a) = 0.7, P(b) = 0.3$. A non-negative square matrix M is *primitive* if for some $d > 0$, the matrix M^d is positive. Call a finite set of words W *primitive* if the sum of the matrices of words from W is primitive. It is well known that if $\mathscr{A}$ is strongly connected and synchronizing, then the matrix of the underlying digraph of $\mathscr{A}$ is primitive, and so is the matrix of a Markov chain of $\mathscr{A}$ for any positive probability distribution P (see e.g. [1,2,7]).

Proposition 1. *Let M be a row stochastic $n\times n$ matrix. Then there exists a* stationary distribution *$\alpha \in \mathbb{R}^n$, that is, a non-negative stochastic vector satisfying $\alpha M = \alpha$. Moreover, if M is primitive then α is unique and positive.*

Call a set of words $W \subseteq \Sigma^*$ *complete* for a subspace $V \leq \mathbb{R}^n$, with respect to a vector $g \in V$, if

$$\langle g[w] \mid w \in W\rangle = V.$$

For a subset $S \subseteq Q$ we define $V_S = \langle [p] \mid p \in S\rangle \leq \mathbb{R}^n$.

We aim to strengthen [7, Theorem 9]. Namely, we show that the condition that $\mathscr{A}$ is synchronizing is not necessary if we require completeness for the corresponding set of words, and that only completeness with respect to the stationary distribution of $\mathscr{A}$ is required. As in [7] we construct an auxiliary automaton. We fix two positive integers d_1, d_2 and two non-empty sets of words $W_1 \subseteq \Sigma^{\leq d_1}$, $W_2 \subseteq \Sigma^{\leq d_2}$. Consider the automaton

$$\mathscr{A}_c(W_1, W_2) = (R, W_2W_1, \delta_{\mathscr{A}_c}),$$

where $R = \{q.w \mid q \in Q,\ w \in W_1\}$ and $W_2W_1 = \{w_2w_1 \in \Sigma^* \mid w_2 \in W_2, w_1 \in W_1\}$. The transition function $\delta_{\mathscr{A}_c}$ is defined in compliance with the actions of words in $\mathscr{A}$, i.e. $\delta_{\mathscr{A}_c}(q, w) = \delta(q, w)$, for all $q \in R$ and $w \in W_2W_1$. Note that $\delta_{\mathscr{A}_c}$ is well defined because $q.w \in R$ for all $q \in Q$ and $w \in \Sigma_{\mathscr{A}_c}$. Without loss of generality we may assume that $R = \{1, 2, \ldots, |R|\}$ where $r = |R|$.

Let P_1 and P_2 be some positive probability distributions on the sets W_1 and W_2, respectively, and denote $[P_i] = \sum_{w\in W_i} P_i(w)[w]$ for $i = 1, 2$. Then the $r \times r$ submatrix formed by the first r rows and the first r columns of the matrix

$$S(\mathscr{B}, P_2P_1) = [P_2][P_1] = \sum_{w_1\in W_1, w_2\in W_2} P_1(w_1)P_2(w_2)[w_2][w_1]$$

is the transition matrix of the Markov chain on $\mathscr{A}_c$. By Proposition 1 there exists a steady state distribution $\alpha = \alpha(\mathscr{A}_c) \in V_R$, that is, a stochastic vector (with first r non-negative entries) satisfying $\alpha S(\mathscr{A}_c, P_2P_1) = \alpha$.

For a vector $g \in \mathbb{R}^n_+$, by $\mathrm{DS}(g)$ we denote the number of different positive sums of entries of g, i.e.

$$\mathrm{DS}(g) = |\{(g, z) \mid z \in \{0,1\}^n\}| - 1.$$

Theorem 1. *Let $\mathscr{A} = (Q, \Sigma, \delta)$ be an automaton and let*

$$\mathscr{B} = \mathscr{A}_c(W_1, W_2) = (R, W_2W_1, \delta_{\mathscr{B}}),$$

be the automaton defined as above. If W_2W_1 is complete for V_R with respect to α, and $w_0 \in \Sigma^*$ is a word with $Q.w_0 = R$, then:

1. *If $x \in V_R \setminus \langle [R] \rangle$, then there exists $w \in W_2W_1$ such that $(x, \alpha[w]) > (x, \alpha)$;*
2. *$\mathscr{B}$ is synchronizing and $\mathrm{rt}(\mathscr{B}) \leq \mathrm{DS}(\alpha) - 1$;*
3. *$\mathscr{A}$ is synchronizing and*

$$\mathrm{rt}(\mathscr{A}) \leq \begin{cases} |w_0| + \mathrm{rt}(\mathscr{B})(d_1 + d_2) \leq |w_0| + (\mathrm{DS}(\alpha) - 1)(d_1 + d_2) & \text{if } R \neq Q, \\ 1 + (\mathrm{DS}(\alpha) - 2)(d_1 + d_2) & \text{if } R = Q. \end{cases}$$

Proof. Let $x \in V_R \setminus \langle [R] \rangle$. We have

$$(x, [q]) \neq (x, \alpha) \text{ for some } q \in R. \tag{1}$$

Since $[q] \in V_R$ and W_2W_1 is complete for V_R with respect to α, we can represent it as follows:

$$[q] = \sum_{w_1 \in W_1, w_2 \in W_2} \lambda_{w_1,w_2} \alpha[w_2][w_1] \text{ for some } \lambda_{w_1,w_2} \in \mathbb{R}. \tag{2}$$

Multiplying (2) by the vector $[Q]$ we obtain

$$1 = ([q], [Q]) = \left(\sum_{w_1 \in W_1, w_2 \in W_2} \lambda_{w_1,w_2} \alpha[w_2][w_1], [Q] \right) = \sum_{w_1 \in W_1, w_2 \in W_2} \lambda_{w_1,w_2}. \tag{3}$$

Multiplying (2) by the vector x we obtain

$$([q], x) = \left(\sum_{w_1 \in W_1, w_2 \in W_2} \lambda_{w_1,w_2} \alpha[w_2][w_1], x \right). \tag{4}$$

Arguing by contradiction, suppose $(x, \alpha[u_2][u_1]) = (x, \alpha)$ for every $u_1 \in W_1$, $u_2 \in W_2$. Then by (3) and (4) we get that $([q], x) = (x, \alpha)$ contradicts (1). Hence

$$(x, \alpha[u_2][u_1]) \neq (x, \alpha),$$

for some $u_1 \in W_1$, $u_2 \in W_2$.

Since $\alpha[P_2][P_1] = \alpha$, we have either $(x, \alpha[u_2][u_1]) > (x, \alpha)$ or $(x, \alpha[v_2][v_1]) > (x, \alpha)$ for some other $v_1 \in W_1, v_2 \in W_2$. Thus Claim 1 follows.

The proof of Claims 2 and 3 follows from an application of the *greedy extension algorithm* from Section 3. □

The following properties are easily verified and will be useful:

Remark 1. If W_2 is complete for $\mathbb{R}^n$ with respect to some vector g, then W_2W_1 is complete for V_R with respect to g.

Remark 2. If for some positive probability distributions on W_2 and W_1, the set W_2W_1 is complete for V_R with respect to each stationary distribution, then $\mathscr{B} = \mathscr{A}_c(W_1, W_2)$ is strongly connected and synchronizing.

Remark 3. If $\mathscr{B} = \mathscr{A}_c(W_1, W_2)$ is strongly connected and W_2W_1 is complete for V_R with respect to a stationary distribution induced by some positive probability distributions on W_2 and W_1, then W_2W_1 is complete for V_R with respect to any stochastic vector.

Criterion 1 *Let α be a stationary distribution of the Markov chain associated with a strongly connected n-state automaton $\mathscr{A}$ by a given positive probability distribution P on the alphabet Σ. Then $\mathscr{A}$ is synchronizing if and only if there exists a set of words W which is complete for $\mathbb{R}^n$ with respect to α.*

Proof. If $\mathscr{A}$ is synchronizing then for each state $q \in Q$ there is a reset word w_q such that $Q.w_q = q$. Hence, $W = \{w_q \mid q \in Q\}$ is complete for $\mathbb{R}^n$ with respect to α, because $\alpha[w_q] = [q]$.

Let us prove the opposite direction. Set

$$W_1 = \{\varepsilon\},\ W_2 = \Sigma^{\leq n-1},\ \text{and } [P_2] = \frac{1}{n}\sum_{i=0}^{n-1}[P]^i.$$

Then $\alpha[P_2] = \alpha$, and W_2 is complete for $\mathbb{R}^n$ with respect to α. Hence $\mathscr{A}$ is synchronizing by Theorem 1. □

It is worth mentioning that an equivalent criterion with respect to our case has been independently obtained in [32] in terms of affine operators via a so called *fixed point* approach.

Now we can provide an upper bound for the reset threshold, if we can find a short word of a small rank.

Theorem 2. *Let $\mathscr{A} = (Q, \Sigma, \delta)$ be a synchronizing automaton. Then there is a unique (strongly connected)* sink component *$\mathscr{S} = (S, \Sigma, \delta)$. Let w be a word and denote $r = |Q.w|$. Let $0 < d < n$ be the smallest positive integer such that $\Sigma^{\leq d}$ is complete for V_S with respect to any stochastic vector $g \in V_S$ and for each $q \in Q$ there is a word $u_q \in \Sigma^{\leq d}$ such that $q.u_q \in S \cap Q.w$. Then*

$$\mathrm{rt}(\mathscr{A}) \leq \begin{cases} (|w| + d)(\frac{r^3 - r}{6}) - d & \text{if } r \geq 4; \\ |w| + (|w| + d)(r-1)^2 & \text{if } r \leq 3. \end{cases}$$

Moreover, any pair of states from Q is compressible by a word of length at most $|w| + (|w| + d)\frac{r^2 - r}{2}$.

Proof. Let $W_1 = \{w\}$, $W_2 = \Sigma^{\leq d}$, $w_0 = w$, and let P_1, P_2 be arbitrary positive distributions on W_1 and W_2, respectively. We define $\mathscr{B} = \mathscr{A}_c(W_1, W_2)$ as in Theorem 1, and consider its sink component $\mathscr{C} = \mathscr{S}_c(W_1, W_2) = (Q_C, \Sigma, W_2W_1)$. Clearly $Q_C = Q.w \cap S$, and W_2W_1 is complete for $V_{Q_C} \leq V_S$ with respect to any stochastic vector $g \in V_{Q_C}$. By Criterion 1 we obtain that $\mathscr{C}$ is synchronizing.

Since for each $q \in Q.w$ there is a word $u_q \in W_2$ and so $w_q \in W_2W_1$ (a letter of $\mathscr{B}$) which takes q to Q_C, the automaton $\mathscr{B}$ is synchronizing.

Since $\mathscr{B}$ is synchronizing, $|Q.w_0| = r$, and $|u| \leq |w| + d$ for each $u \in W_2W_1$, we have that $\mathrm{rt}(\mathscr{A}) \leq |w| + \mathrm{rt}(\mathscr{B})(|w| + d)$. By Pin's bound for the reset threshold in the general case [28], $\mathrm{rt}(\mathscr{B}) \leq \frac{r^3-r}{6} - 1$ for $r \geq 4$.

Since $\mathscr{B}$ is synchronizing and there are $\frac{r^2-r}{2}$ pairs in $Q.w$, any pair of states in Q can be compressed by a word of length at most $|w| + (|w| + d)\frac{r^2-r}{2}$. □

3 Finding reset words of lengths within the bounds

Throughout this section suppose we are given a strongly connected automaton $\mathscr{A}$, a word w_0 such that $Q.w_0 = R$ for some $R \subseteq Q$, a non-empty polynomial set of words W_1 with a positive distribution P_1, and a set of words W_2 with a positive distribution P_2, which satisfy Theorem 1.

Consider the case when W_2 is of polynomial size. Then we can calculate the dominant eigenvector $\alpha \in \mathbb{R}^n$ of the matrix $[P_2][P_1]$. Under certain assumptions on rationality of the distributions, it can be done in polynomial time. Next, depending on whether the bound is obtained by Theorem 2 or Claim 2 of Theorem 1, we use either a greedy compressing algorithm (such as in [16]), or the following *greedy extension algorithm*, respectively.

The Greedy Extension Algorithm. We start from $x_0 = [q]$ for $q \in R$ and by Claim 2 of Theorem 1 find $u_0 \in W_2W_1$ such that $(x_0, \alpha[u_0]) > (x_0, \alpha)$. For $i = 0, 1, \ldots$ following this way until $x_i \in \langle [R] \rangle$, find for $x_{i+1} = x_i[u_i]^t$ a word $u_{i+1} \in W_2W_1$ such that $(x_{i+1}, \alpha[u_{i+1}]) > (x_{i+1}, \alpha)$. Since x_i is a 1-0 vector, we need at most $\mathrm{DS}(\alpha) - 1$ steps until $x_i = [q]([u_iu_{i-1} \ldots u_0])^t = [R]$. As the result we return the word $w_0u_iu_{i-1} \ldots u_0$. Notice that in the case when $R = Q$ we can choose q such that for some letter $a \in \Sigma$, we have $|q.a^{-1}| > 1$ and set $u_0 = a$. □

The problem is that usually W_2 is given by $\Sigma^{\leq d}$ for some $d = \mathrm{poly}(n)$. The following reduction procedure allows to replace potentially exponential set W_2 with a polynomial set of words W, whose the longest words are not longer than those of W_2.

The Reduction Procedure. The procedure takes a number d, and returns a polynomial subset $W \subseteq \Sigma^{\leq d}$ such that $\langle W \rangle = \langle \Sigma^{\leq d} \rangle$ and the maximum length of words from W is the shortest possible.

We start with $V_0 = \{I_n\}$ and $W = \{\varepsilon\}$. In each iteration $i \in \{1, 2, \ldots\}$ we first set $V_{i+1} = V_i$. Then we subsequently check each letter $a \in \Sigma$ and each word $u \in W$ of length i: If the matrix $[ua]$ does not belong to the subspace V_{i+1}, we add the word ua to W and the matrix $[ua]$ to the basis of V_{i+1}. We stop the procedure at the first iteration where nothing is added.

Since in an i-th iteration we have considered $a \in \Sigma$ and $u \in W$ of length less than i in the previous iterations, by induction we get

$$V_i = \langle I_n(W \cap \Sigma^{\leq i}) \rangle = \langle I_n\Sigma^{\leq i} \rangle.$$

It follows from the ascending chain argument (see e.g. [30,21]) that for some $j < n$ we have

$$V_j = V_{j+1} = \ldots.$$

Thus the procedure is stopped at the first such j, and $j \leq \min\{d, n-1\}$. We get that $\langle W \rangle = V_j = \langle \Sigma^d \rangle$. Since in each step we add only independent matrices as the basis of V_{i+1}, we get $|W| = \dim(V_j)$. Also the lengths of words in W are at most $j \leq \min\{d, n-1\}$. □

Using the reduction procedure for total completeness we can replace Σ^d from Theorem 2 by a polynomial W, which is also complete for V_S with respect to any stochastic vector $g \in V_S$.

Hence, this yields a polynomial time algorithm finding reset words of lengths within the bound of Theorem 2.

In some situations we are interested only in completeness with respect to a given vector α. Then we can find a reduced set W of potentially shorter words than that obtained by the general reduction procedure.

The Reduction Procedure for α-Completeness. The procedure takes a number d and a vector α, and returns a polynomial subset $W \subseteq \Sigma^{\leq d}$ such that $\langle \alpha W \rangle = \langle \alpha \Sigma^{\leq d} \rangle$ and the maximum length of words from W is the shortest possible.

We just follow the general reduction procedure, where instead of matrix spaces we consider vector spaces. It is enough to replace I_0 by α, and we obtain $\langle \alpha W \rangle = V_j = \langle \alpha \Sigma^{\leq d} \rangle$. □

Remark 4. Instead of $\Sigma^{\leq d}$ the reduction procedures can also reduce any set of words $W' \subset \Sigma^*$ that is factor-closed. A set of words W' is *factor-closed* if $uvw \in W'$ implies that $uw \in W'$, for each $u, v, w \in \Sigma^*$.

This follows since the ascending chain argument still holds. If $V_i = V_{i+1}$ then also $V_i = \langle W' \rangle$. Assume for a contrary that $V_i < \langle W' \rangle$, and let $ua \in W'$ be a shortest word such that $I_n[ua]$ is independent to $I_n W$. Then there is some prefix $w \in W$ of u, and $u = wva$. Since u was a shortest word, wv is dependent to $I_n W$, so

$$\langle I_n W \rangle = \langle I_n(W \cup [wv]) \rangle < \langle I_n(W \cup [wva]) = \langle I_n(W \cup [wa]).$$

Since W' is factor-closed, $wa \in W'$, and it was considered in the reduction procedure and added to W – a contradiction. □

The following procedure finds a polynomial subset $W \subseteq W_2$ such that WW_1 is still primitive under the restriction to V_R, and the words in W are as short as possible.

The Reduction Procedure for Primitive Sets. As the input the procedure takes a set of words W_1 and a number $d > 0$ such that $\Sigma^{\leq d} W_1$ is primitive when restricted to V_R where $R = Q.W_1$, and returns a polynomial subset $W \subseteq \Sigma^{\leq d}$ such that WW_1 is also primitive for R.

We follow the reduction procedure with the following modification: Instead of adding a word ua to W if $[ua]$ does not belong to the current subspace V_{i+1}, we add ua if for some $w_1 \in W_1$ there is a non-zero entry in $[ua][w_1]$ such that this entry is zero in all matrices $[w]$ for $w \in WW_1$. We stop the procedure as soon as the set of words WW_1 restricted to V_R becomes primitive.

To check whether WW_1 is primitive, since the exponent of $r \times r$ primitive matrix is at most $(r-1)^2+1$ (see [2]), it is enough to check that the $((r-1)^2+1)$-th power of the sum of all matrices $[w]$ for $w \in WW_1$ is positive. Since in each step we make positive at least one entry in this sum, we need at most $(r-1)^2+1$ steps in total. □

Now, given some sets W_1 and $W_2 = \Sigma^{\leq d}$, we can first find $W \subseteq W_2$ such that WW_1 is primitive for V_R. Then, we can choose some positive probability distribution on W, which induces a unique stationary distribution β. We can also find $W' \subseteq W_2$ complete with respect to β. The problem here is that, for the set $(W \cup W')W_1$ there is possibly no positive probability distribution inducing the stationary distribution β. In order to apply Theorem 1, we need to show that $(W \cup W')W_1$ can be complete with respect to its stationary distribution. The following theorem solves this problem.

Theorem 3. *Let $\mathscr{A}$ be a strongly connected automaton, and the sets W_1, W_2 be chosen so that the matrix of the underlying digraph of $\mathscr{B} = \mathscr{A}_c(W_1, W_2)$ is primitive. Let α be a stationary distribution of the Markov chain associated with $\mathscr{B}$ for arbitrary positive distributions P_1, P_2 on W_1, W_2, respectively. Then, for each set of words W which is complete for $\mathbb{R}^n$ with respect to α, the automaton $\mathscr{C} = \mathscr{A}_c(W_1, W \cup W_2)$ is synchronizing.*

Proof. For each $0 \leq \delta < 1$ we define

$$S(\delta) = ((1-\delta)[P_2] + \frac{\delta}{|W|}\sum_{w\in W}[w])[P_1].$$

Clearly, $S(\delta)$ is a positive probability distribution on $(W \cup W_2)W_1$ for each $0 < \delta < 1$, and on W_2W_1 for $\delta = 0$. Because the matrix of the underlying digraph of $\mathscr{B}$ is primitive, for each $0 \leq \delta < 1$ there is a unique stationary distribution $\beta(\delta)$ such that $\beta(\delta)S(\delta) = \beta(\delta)$ or, equivalently, $\beta(\delta)$ is the unique stochastic solution x of the equation

$$x(S(\delta) - I_n) = (0, 0, \ldots, 0).$$

Therefore $\beta(\delta) = \tilde{S}^{-1}(\delta)(1, 0, \ldots, 0)$, where $\tilde{S}(\delta)$ is the invertible matrix obtained from the matrix $S(\delta) - I_n$ by replacing the first row by the vector of all 1-s. Note that $\beta(0) = \alpha$, and $\beta(\delta)$ is (component wise) continuous in $[0, 1)$.

Since W is complete with respect to α, there are words $w_1, w_2, \ldots, w_n \in W$ such that the square matrix $D = (\alpha w_i)_{i\in\{1,2,\ldots,n\}}$ has rank n. For $0 \leq \delta < 1$ define the matrix

$$D_\delta = (\beta(\delta)w_i)_{i\in\{1,2,\ldots,n\}}$$

and consider the function $\phi(\delta) = \det(D_\delta)$. Since $\beta(\delta)$ is continuous in $[0, 1)$, $\phi(\delta)$ is also continuous in $[0, 1)$. Since $\phi(0) = \det(D) \neq 0$, we get that $\phi(\delta') \neq 0$ for some $0 < \delta' < 1$. Hence W is complete for $\mathbb{R}^n$ with respect to $\beta(\delta')$. Since $\beta(\delta')$ is the stationary distribution of the Markov chain defined on $(W \cup W_2)W_1$ by the positive probability distribution $S(\delta')$, by Theorem 1 we obtain that the automaton $\mathscr{C}$ is synchronizing. □

3.1 Synchronizing quasi-Eulerian automata

Let α be the probability distribution on $\Sigma^{\leq d}$ induced by a probability distribution $P\colon \Sigma \mapsto \mathbb{R}^+$ on the alphabet, that is, $[P_2] = \frac{1}{n}\sum_{i=0}^{d}[P]^i$. Suppose that $d < \mathrm{poly}(n)$ is such that $\Sigma^{\leq d}$ is complete for $\mathbb{R}^n$ with respect to α. Using the reduction procedure, we can construct a set U of at most n words such that

$$\langle \alpha U\rangle = \langle \alpha\Sigma^{\leq d}\rangle = \mathbb{R}^n.$$

However, α is not necessarily the stationary distribution for some positive probability distribution on U. The following lemma solves this problem.

Lemma 1. *Let $W = \{au \mid u \in \mathrm{Suff}(U), a \in \Sigma\}$, where $\mathrm{Suff}(U)$ is the set of proper suffixes of U. Then there exists a positive probability distribution on W such that α is the corresponding stationary distribution.*

Proof. Since W is complete with respect to α, following the proof of Theorem 1 for each $x \in \mathbb{R}^n \setminus \langle [Q]\rangle$, there exists $w \in W$ such that $(x, \alpha[w]) \neq (x, \alpha)$. Suppose that w is a shortest word from W with this property. If $(x, \alpha[w]) > (x, \alpha)$ then we have found an extension word from W. Suppose that $(x, \alpha[w]) < (x, \alpha)$. Clearly $1 \leq |w| \leq d$, and $w = au$ for $a \in \Sigma$ and $u \in \Sigma^{\leq d-1}$. Since

$$(x, \alpha[u]) = (x, \alpha[P][u]) = P(a)(x, \alpha[w]) + \sum_{b\in\Sigma, b\neq a} P(b)(x, \alpha[bu]),$$

we get that either $(x, \alpha[u]) < (x, \alpha)$ or $(x, \alpha[bu]) > (x, \alpha)$ for some $b \neq a$. Since $w \in W$, we have $u \in \mathrm{Suff}(U)$ and so $bu \in W$. If $(x, \alpha[u]) < (x, \alpha)$ then $u \neq \varepsilon$, so $u \in W$, and u is a shorter word with $(x, \alpha[u]) \neq (x, \alpha)$, which contradicts the choice of w. Therefore by [7, Theorem 13] the automaton $\mathscr{B} = \mathscr{A}_c(\{\varepsilon\}, W)$ is synchronizing and α is the stationary distribution for some probability distribution on W. □

As an application we get a polynomial algorithm for finding a reset word for the class of *quasi-Eulerian* automata, a generalization of Eulerian automata. We call an automaton $\mathscr{A}$ *quasi-Eulerian* with respect to an integer $c \geq 0$ if it satisfies the following two conditions:

1. There is a subset $E_c \subseteq Q$ containing $n - c$ states such that only one of these states, say s, can have incoming edges from the set $Q \setminus E_c$.
2. There exists a positive probability distribution P on Σ such that the columns of the matrix $[P]$ that correspond to the states from $E_c \setminus \{s\}$ sum up to 1.

Within this definition, for $c = 0$ we get so-called *pseudo-Eulerian* automata, and if additionally P is uniform on Σ, then we get Eulerian automata. The upper bound $1 + (n-2)(n-1)$ on the reset thresholds of Eulerian automata was found by Kari [21], and extended to the class of pseudo-Eulerian automata by Steinberg [29]. These results were generalized in [7, Corollary 11] by showing the upper bound $2^c(n-c+1)(n-1)$ for the class of quasi-Eulerian automata with respect to a non-negative integer c. The following theorem gives a polynomial time algorithm for finding reset words satisfying these bounds.

Theorem 4. *Given a synchronizing automaton $\mathscr{A}$ which is quasi-Eulerian with respect to an integer $c \geq 0$, there is a polynomial time algorithm for finding a reset word of length at most:*

$$\begin{cases} 2^c(n-c+1)d & \text{if } c > 0; \\ 1 + (n-2)d & \text{if } c = 0, \end{cases}$$

where $d \leq n-1$ is the smallest integer such that $\Sigma^{\leq d}$ is complete.

Proof. First we need to calculate a stationary distribution α, which has $n - c$ equal entries. For this purpose, for each of the $\binom{n}{n-c}$ ways of choosing the set E_c containing $n - c$ states, we find a solution of the following task of linear programming:

$$\begin{cases} \alpha[P] = \alpha, \\ ([Q], \alpha) = 1, \\ \alpha_p = \alpha_q & \text{for each } p \in E_c, \\ P(a) > 0 & \text{for each } a \in \Sigma; \end{cases}$$

with the variable set

$$\{P(a) \mid a \in \Sigma\}, \{\alpha_p \mid p \in Q\},$$

and q is an arbitrary state from E_c. If there is a solution (α, P), then α is the stationary distribution for the positive probability distribution P on the alphabet and it has at least $n - c$ equal entries. Since $\binom{n}{n-c}$ is polynomial and linear programming is solvable in polynomial time, such a solution can be found in polynomial time.

Next, according to the reduction procedure for α-completeness we can find a polynomial set of words $W' \subseteq \Sigma^{\leq d}$ which is complete for $\mathbb{R}^n$ with respect to α. Due to Lemma 1 we can change the set W' to a set W of polynomial size preserving the stationary distribution α and then use the greedy extension algorithm to find a reset word of the proposed lengths. □

3.2 Synchronizing quasi-one-cluster automata

The *underlying digraph* of a letter $a \in \Sigma$ is the digraph with edges labeled by a. Every connected component, called *cluster*, in the underlying digraph of a letter has exactly one cycle, and possible some trees rooted on this cycle. An automaton $\mathscr{A} = (Q, \Sigma, \delta)$ is called *one-cluster* if there is a letter $a \in \Sigma$ whose underlying digraph has only one cluster. An automaton $\mathscr{A}$ is *quasi-one-cluster* with respect to an integer $c \geq 0$ if it has a letter whose underlying digraph has a cluster such that there are at most c states in the cycles of all other clusters. Clearly, one-cluster automata are quasi-one-cluster with respect to $c = 0$. An automaton $\mathscr{A}$ is *circular* is it has a letter whose underlying digraph consists of only one cycle of length n.

The Černý conjecture was proved for *circular* automata [15], and for one-cluster automata with prime length cycle [30]. Also, quadratic bounds for the reset thresholds in the general case of one-cluster automata were presented [4,3,29,12]. In [7] the upper bound $2^c(2n - c - 2)(n - c + 1)$ was proved for quasi-one-cluster with respect to c.

The following theorem gives a polynomial algorithm finding a reset word for quasi-one-cluster automata, whose length is of the mentioned bounds.

Theorem 5. *Let $\mathscr{A}$ be a synchronizing automaton that is quasi-one-cluster with respect to a letter a and $c \geq 0$. Let C be the largest cycle of a and h be the maximal height of the trees labeled by a. Let $W_1 = \{a^{h+i} \mid i \in \{0, \ldots, |C| - 1\}\}$. Then there is a polynomial algorithm for finding a reset word for $\mathscr{A}$ of length at most*

$$\begin{cases} 2^c(2n - c)(n - c + 1) & \text{if } c > 0; \\ 1 + (2n - r)(n - 2) & \text{if } c = 0, \end{cases}$$

where r is the smallest dimension of $\langle W_1\beta \rangle$ for $\beta \in V_C \setminus \langle [C] \rangle$. In particular, if $|C|$ is prime then $r = |C|$.

Proof. We can assume that $\mathscr{A}$ is strongly connected; otherwise, we can use the same technique as in Theorem 2.

Let us define $W_2 = \Sigma^{\leq n-r+1}$ for $c = 0$ and $W_2 = \Sigma^{\leq n-1}$ otherwise. It is proved in [30] that for one-cluster automata each non-trivial subset of $S \subseteq C$ can be extended to a bigger one by a word from W_2W_1. Hence due to the greedy extension algorithm the induced automaton is synchronizing and W_2W_1 is complete for V_C with respect to any stochastic vector from V_C. Thus in both cases we get that W_2W_1 is complete for $V_{Q.a^h}$. Using the reduction procedure W_2 can be replaced with a polynomial set of words W while keeping the maximal length of words.

Let β be the stationary distribution for some positive distribution on W_2W_1. Then $\beta_p > 0$ if and only if p is a cycle state and $\beta_p = \beta_q$ for each $p, q \in C$. Clearly $\mathrm{DS}(\beta) \leq 2^c(|C| + 1)$ if $c > 0$, and $\mathrm{DS}(\beta) = |C| - 1$ if $c = 0$. According to Theorem 1 the automaton $\mathscr{B} = \mathscr{A}_c(W_1, W)$ is synchronizing and we get that

$$\mathrm{rt}(\mathscr{A}) \leq \begin{cases} h + 2^c(|C| + 1)(h + |C| + n) & \text{if } c > 0; \\ 1 + (h + |C| + n - r)(n - 2) & \text{if } c = 0. \end{cases}$$

Since the worst case appears when $|C| = n - c$ and $h = 0$, the bound follows. Since W_1 and W have polynomial size, a reset word of this bound can be found by the greedy extension algorithm in polynomial time. □

Remark 5. The algorithm of Theorem 5 works also for the bounds from [12] for one-cluster automata. This follows in the same way as referring to [30] in the proof of the theorem.

4 The Černý conjecture and random automata

Using the new bound, we can extend the class of automata for which the Černý conjecture is proven. In particular, we can improve the result from [27], where the Černý conjecture is proven for automata with a letter of rank at most $1 + \log_2 n$.

Corollary 1. *Let $\mathscr{A} = (Q, \Sigma, \delta)$ be a synchronizing automaton. If there is a letter of rank*

$$r \leq \sqrt[3]{6n-6},$$

then $\mathscr{A}$ satisfies the Černý conjecture.

Proof. Assume that $r \geq 3$. Using Theorem 2 with $d = n - 1$ and $|w| = 1$ we obtain the bound $\mathrm{rt}(\mathscr{A}) \leq n(\frac{r^3-r}{6} - 1) + 1$. Then using $r \leq \sqrt[3]{6n-6}$ we obtain

$$\mathrm{rt}(\mathscr{A}) < n\left(\frac{r^3}{6} - 1\right) + 1 \leq n\left(\frac{6n-6}{6} - 1\right) + 1 = (n-1)^2.$$

If $r \leq 3$ then the bound of Theorem 2 is $1 + n(r-1)^2$, which is not larger than $(n-1)^2$ for $n \geq 6$. For $n \leq 5$ the Černý conjecture has been verified [23]. □

Another corollary concerns random synchronizing automata. We consider the uniform distribution P_s on all synchronizing binary automata with n states, which is formally defined by $P_s(\mathcal{A}) = P(\mathcal{A})/P_n$, where P is the uniform distribution on all n^{2n} binary automata, and P_n is the probability that a uniformly random binary automaton is synchronizing. It is known that P_n tends to 1 as n goes to infinity ([6,25]).

Given an arbitrary small $\varepsilon > 0$ and n large enough, Nicaud [25] proved that a random binary automaton with n states has a word of

1. length $n^{3/4+3\varepsilon}(1 + o(1))$ and rank at most $n^{1/4+2\varepsilon}$ with probability at least $1 - O(\exp(-n^{\varepsilon}/4))$,
2. length $n^{7/8+7\varepsilon}(1 + o(1))$ and rank at most $n^{1/8+4\varepsilon}$ with probability at least $1 - O(n^{-1/4+3\varepsilon})$,

The following corollary is a consequence of these results and our Theorem 2.

Corollary 2. *For any $\varepsilon > 0$ and n large enough, with probability at least $1 - O(\exp(n^{-\varepsilon/4}))$, a random n-state automaton with at least two letters has a reset word of length at most $n^{7/4+6\varepsilon}(1 + o(1))$, and so satisfies the Černý conjecture. Moreover, the expected value of the reset threshold of a random synchronizing binary automaton is at most $n^{3/2+o(1)}$.*

Proof. Because a random binary automaton is synchronizing with high probability, the probabilities in (1) and (2) remain asymptotically at least the same for a random binary synchronizing automaton.

Now, by applying our Theorem 2 (with $d = n - 1$) to (1) and (2) we get that a random binary synchronizing automaton has a reset word of

1. length $n^{7/4+6\varepsilon}(1 + o(1))$ with probability at least $1 - O(\exp(-n^{\varepsilon}/4))$,
2. length $n^{11/8+12\varepsilon}(1 + o(1))$ with probability at least $1 - O(n^{-1/4+3\varepsilon})$.

Claim (1) is the first statement of the corollary.

Calculating an upper bound of the average of (1), (2) and the general cubic bound applied to the rest of automata we get:

$$\begin{aligned} & n^{11/8+12\varepsilon}(1+o(1)) + \\ & n^{7/4+6\varepsilon}(1+o(1)) \cdot O(n^{-1/4+3\varepsilon}) + \\ & (n^3-n)/6 \cdot O(\exp(-n^{\varepsilon}/4)) \\ = {} & O(n^{11/8+12\varepsilon}) + O(n^{6/4+9\varepsilon}) \\ = {} & O(n^{3/2+12\varepsilon}). \end{aligned}$$

□

5 Synchronizing finite prefix codes

A *finite prefix code* (Huffman code) $\mathcal{T}$ is a set of N ($N > 0$) non-empty words $\{w_1, \ldots, w_N\}$ from Σ^*, such that no word in $\mathcal{T}$ is a prefix of another word in $\mathcal{T}$. A finite prefix code $\mathcal{T}$ is *maximal* if adding any word $w \in \Sigma^*$ to $\mathcal{T}$ does not result in a finite prefix code. We consider only maximal prefix codes. A *reset word* for the code $\mathcal{T}$ is a word w such that for any $u \in \Sigma^*$ the word uw is a sequence of words from $\mathcal{T}$.

One can easily see that a finite prefix code corresponds naturally to a DFA called the *decoder*, whose states are proper prefixes of words from this code [11]. Formally, for a finite prefix code $\mathcal{T}$ we have the corresponding *decoder* $\mathscr{A}_{\mathcal{T}}$, which is the DFA (Q, Σ, δ) with $Q = \{q_v \mid v \text{ is a proper prefix of a word in } \mathcal{T}\}$, and δ defined as follows:

$$\delta(q_v, a) = \begin{cases} q_{va} & \text{if } va \notin \mathcal{T}; \\ q_\varepsilon & \text{otherwise.} \end{cases}$$

If for an edge from a state q_v to the root q_ε we assign an output symbol associated with the word q_v, the decoder can read a compressed input string and produce the decompressed output according to the code $\mathcal{T}$. Observe that a reset word w for $\mathcal{T}$ is a reset word for the decoder $\mathscr{A}_{\mathcal{T}}$, and $Q.w = \{q_\varepsilon\}$. The decoder $\mathscr{A}_{\mathcal{T}}$ naturally corresponds to a rooted k-ary tree. We say that q_ε is the *root* state. The *level* of a state $q_v \in Q$ is $|v|$, which is also the length of the shortest path from q_ε to q_v in the decoder DFA. The *height* of $\mathscr{A}_{\mathcal{T}}$ is the maximal level of the states in Q; this is also the maximal length of words from $\mathcal{T}$.

Remark 6. If $N = |\mathcal{T}|$ and $k = |\Sigma|$, then the number n of states of $\mathscr{A}_{\mathcal{T}}$ is $\frac{N-1}{k-1}$. Note that it does not depend on the length of the words in the code.

In [10,11] Biskup and Plandowski gave an $O(nh \log n)$ upper bound for the reset thresholds of binary decoders, where h is the maximum length of a word from the code. Since h can be linear in terms of n, this is an $O(n^2 \log n)$ general bound. Later, it was improved to $O(n^2)$ in [3]. However, in the worst case, only decoders with a reset threshold in $\Theta(n)$ are known [11], and it was conjectured that every synchronizing decoder possess a synchronizing word of length $O(n)$. Thus, there was a big gap between the upper and lower bounds for the worst case.

The following lemma is a simple generalization of [11, Lemma 14] to k-ary decoders.

Lemma 2. *Let $\mathscr{A}_\mathcal{T} = (Q, \Sigma, \delta)$ be the n-state k-ary synchronizing decoder of a finite prefix code $\mathcal{T}$. There is a word w of rank $r \leq \lceil \log_k n \rceil$ and length r.*

Proof. For a word w, we define

$$Q(w) = \{q_v.w \mid q_v \in Q \text{ such that no prefix of } w \text{ maps } q_v \text{ to } q_\varepsilon\}.$$

Consider $r > 0$. Observe that for two distinct words w_1, w_2 of the same length r the sets $Q(w_1)$ and $Q(w_2)$ are disjoint. Also the states in $Q(w)$ are of level at least $r+1$. If for all words of length r the sets $Q(w)$ are non-empty, then there are at least k^r states in Q of level at least $r+1$, because there are k^r different words of length r. Then $k^r + r + 1 \leq n$ and $r < \log_k n$. Hence, if $r = \lceil \log_k n \rceil$ then there exists a word w with the empty $Q(w)$. Since any state is mapped to q_ε by a prefix of w, the rank of w is at most $|w| = r$. □

Since there exists a short word of small rank r, we can apply Theorem 2 to improve the general upper bounds for the reset threshold of decoders.

Corollary 3. *Let $\mathscr{A}_\mathcal{T} = (Q, \Sigma, \delta)$ be the n-state k-ary synchronizing decoder of a finite prefix code $\mathcal{T}$, and let $r = \lceil \log_k n \rceil$. Then*

1. $\mathrm{rt}(\mathscr{A}_\mathcal{T}) \leq \begin{cases} 2 + (r+n-1)(\frac{r^3-r}{6} - 1) & \text{if } r \geq 4; \\ 2 + (r+n-1)(r-1)^2 & \text{if } r \leq 3. \end{cases}$
2. *Any pair of states from Q is compressible by a word of length at most*

$$r + (r+n-1)\frac{r^2-r}{2}.$$

Proof. For Claim 1 we apply Theorem 2 with w being the word of rank at most r and length at most r from Lemma 2, and $d = n-1$. This gives $(r+n-1)(\frac{r^3-r}{6}) - (n-1) = r + (r+n-1)(\frac{r^3-r}{6} - 1)$ for $r \geq 4$.

We can slightly refine the bound by Pin's result [27, Proposition 5], which states that if we can compress $Q.w$, then a shortest compressing word for $Q.w$ has length at most $|w| + n - |Q.w| + 1$. Thus if $|Q.w| = r$ this is $n+1$, and we end up with

$$r - (r+n-1) + (n+1) + (r+n-1)\left(\frac{r^3-r}{6} - 1\right) = 2 + (r+n-1)\left(\frac{r^3-r}{6} - 1\right).$$

Similar calculation applies when $r \leq 3$.

Claim 2 follows directly from Theorem 2. □

If the size k of the alphabet is fixed, Corollary 3 yields $O(n \log^3 n)$ upper bound for the reset threshold, and $O(n \log^2 n)$ upper bound for the length of a word compressing a pair of states of a decoder.

Note that the word w from Lemma 2 can be easily computed in $O(n^2)$ time, since there are $O(n)$ words of length at most $\lceil \log_k n \rceil$. Then a reset word within the bound of Corollary 3 can be computed in polynomial time by the algorithm discussed in Section 3.

5.1 Random binary decoders

By a *uniformly random n-state decoder* we understand a decoder chosen uniformly at random from the set of all n-state decoders. We consider here random binary decoders. In [17] it was proved that a uniformly random binary n-state decoder is synchronizing with a probability that tends to 1 as n goes to infinity. Since every binary n-state decoder correspond to a binary tree with $n+1$ leaves, the number of all such decoders is the n-th Catalan number. Note that it is known that the average height of a binary n-state decoder is asymptotically $\Theta(\sqrt{n})$ [24].

Theorem 6. *The expected reset threshold of a uniformly random synchronizing binary n-state decoder is at most $O(n \log n)$.*

Proof. Let $\mathcal{T}$ be the code (set of codewords) of a uniformly random synchronizing binary n-state decoder. Let a be the first letter of the alphabet. First we show that the length of the left-most branch of the decoder is at most logarithmic with high probability. In other words, the unique word from $a^* \cap \mathcal{T}$ has length $\ell \in O(\log n)$ with high probability. Because a uniformly random binary n-state decoder is synchronizing with high probability, these probabilities transfer to a uniformly random synchronizing binary n-state decoder. Note that if a^ℓ is in the code, then the the letter a in the decoder is one-cluster with the cycle of length ℓ. Then we apply the upper bound $O(\ell n)$ on the reset thresholds of one-cluster automata whose the length of the cycle is ℓ [3].

From the proof of [17, Lemma 5.9] we have that the fraction of decoders whose code contains a^ℓ (for $1 \leq \ell \leq n-1$) is equal to:

$$\frac{\ell(n+1)(n)\dots(n-\ell+1)}{(2n-1)(2n-2)\dots(2n-\ell-1)}.$$

Let d be such that $3 \leq d \leq n-1$. Then

$$\frac{n-d+1}{2n-d-1} = \frac{1}{2} - \frac{d-3}{2(2n-d-1)} \leq 1/2$$

Hence for $\ell \geq 3$, we have

$$\frac{\ell(n+1)(n)\dots(n-\ell+1)}{(2n-1)(2n-2)\dots(2n-\ell-1)} \leq \frac{\ell}{2^{\ell+1}}.$$

It follows that the probabilities that a^ℓ is in the code for $\ell \geq 2\log_2 n$ is at most $O(1/n)$.

For the case $\ell < 2\log_2 n$, we use the upper bound $O(\ell n)$ from [3] and for $\ell \geq 2\log_2 n$ with probability $O(1/n)$ we use the general upper bound $O(n\log^3 n)$ for decoders from Corollary 3. Summing up these cases yield our upper bound on the expected reset threshold of the decoder:

$$O(n\log n) + O(n\log^3 n) \cdot O(1/n) = O(n \log n).$$

□

As in the general case, a reset word of average length $O(n \log n)$ can be computed in polynomial time. This can be done using the algorithm discussed in Subsection 3.2, which finds a reset word within the bounds for (quasi-)one-cluster automata.

5.2 Lower bounds

Biskup and Plandowski [10,11] presented a series of binary n-state decoders with the reset threshold $2n-5$ for even n and $2n-7$ for odd n. However, only binary decoders were studied. Here we present a series of k-ary decoders for every $k \geq 3$ with large reset thresholds. This shows that, in the worst case, also for arbitrary large alphabets a decoder can have the reset threshold in $\Theta(n)$.

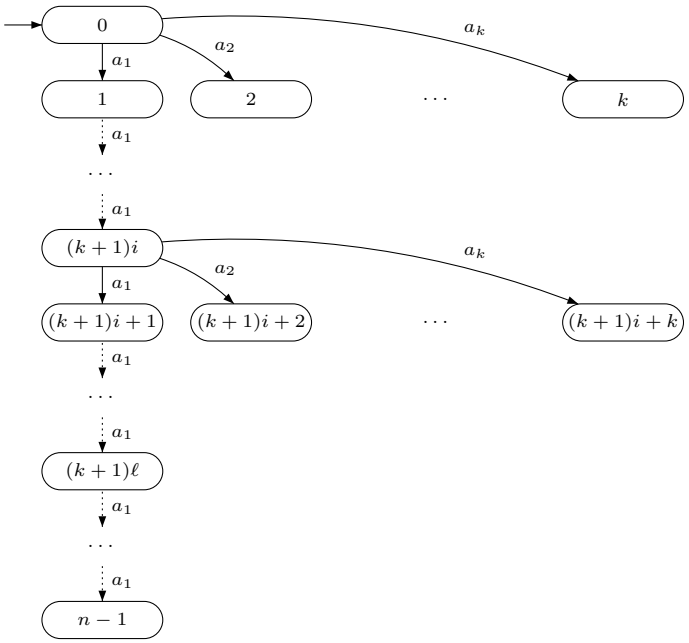

Fig. 2. The decoder $\mathscr{X}_{n,k}$ with reset threshold $\lceil n/(k+1) \rceil$.

For $k \geq 3$ and $n \geq k+2$, we define $\mathscr{X}_{n,k} = (Q, \Sigma, \delta)$ (shown in Fig. 2). Let $Q = \{0, \ldots, n-1\}$ and $\Sigma = \{a_1, \ldots, a_k\}$, and let $\ell = \lceil n/(k+1) \rceil - 1$ (so $\ell \geq 1$). We define δ as follows: For each i with $0 \leq i \leq \ell - 1$ and each $1 \leq j \leq k$, if $(k+1)i+j \leq n-1$ then we define: $\delta((k+1)i+1, a_j) = (k+1)i+j$. Also for i with $(k+1)\ell \leq i \leq n-2$ we define $\delta(i, a_1) = i+1$. For all the remaining states i and letters a_j we set $\delta(i, a_j) = 0$.

Theorem 7. *The automaton $\mathscr{X}_{n,k}$ is synchronizing and its reset threshold is $2\ell+2 = 2\lceil n/(k+1) \rceil$.*

Proof. One verifies that the action of the word $a_k(a_1)^{2\ell}a_k$ synchronizes the automaton.

Let w be a shortest reset word for the automaton. Consider the first two letters w_1, w_2 of w. Observe that $Q.w_1$ and $Q.w_1w_2$ contains 0. So $Q.w_1w_2$ also contains a state p from $\{2, \ldots, k\}$.

State 0 is at the level 0, and state p is at the level 1. For all states $q < (k+1)\ell$ the action of all the letters alternates the parity of the level of the states. Thus two such states with an odd and an even level cannot be compressed by the action of a single letter. So, to compress $\{0, p\}$, one of the states must be first mapped to a state $q \geq (k+1)\ell$. The shortest such a path is from 1 to $(k+1)\ell$

labeled by $(a_1)^{2^\ell-1}$. Then we need one more letter (a_k) to synchronize the pair. It follows that 0 and p requires a word of length at least 2ℓ to be compressed. Hence, the length of w is at least $2+2\ell$. □

Using a more sophisticated construction, it is possible to modify our series and obtain decoders with slightly larger reset thresholds, though still of order $2n/(k+1)+O(1)$. We suppose that this order of growth is tight for $k \geq 3$ up to the constant within $O(1)$.

6 Conclusions and open problems

We have shown constructible upper bounds for the reset threshold, which turned out to be useful in several important cases of automata. They are obtained using a uniform approach basing on Markov chains. In all the cases, there exists a polynomial algorithm finding a reset word of length within the bounds. Also, note that if the Černý conjecture is true, then our bounds become reduced; in particular, we would get $O(n\log^2 n)$ for the reset thresholds of decoders of finite prefix codes.

The questions about tight bounds in the case of finite prefix codes and random automata remain open. For finite prefix codes the bound $O(n)$ was conjectured. Note that for some applications it can be also important to get bounds in terms of the maximal length of the words in the code (e.g. [13]). There is also an interesting question about the expected reset threshold of the decoder of a random finite prefix code. So far for this case we have only the bound $O(n\log n)$, which comes from the bounds for one-cluster automata.

Also, there is the open problem of designing a polynomial algorithm finding reset words within the bound of [15] for circular automata.

Acknowledgments

This work was supported by the Presidential Program "Leading Scientific Schools of the Russian Federation", project no. 5161.2014.1, the Russian Foundation for Basic Research, project no. 13-01-00852, the Ministry of Education and Science of the Russian Federation, project no. 1.1999.2014/K, and the Competitiveness Program of Ural Federal University (Mikhail Berlinkov), and by the National Science Centre, Poland under project number 2014/15/B/ST6/00615 (Marek Szykuła).

References

1. D. S. Ananichev, V. V. Gusev, and M. V. Volkov. Slowly synchronizing automata and digraphs. In *Mathematical Foundations of Computer Science*, volume 6281 of *LNCS*, pages 55–65. Springer, 2010.
2. D. S. Ananichev, M. V. Volkov, and V. V. Gusev. Primitive digraphs with large exponents and slowly synchronizing automata. *Journal of Mathematical Sciences*, 192(3):263–278, 2013.
3. M.-P. Béal, M. V. Berlinkov, and D. Perrin. A quadratic upper bound on the size of a synchronizing word in one-cluster automata. *International Journal of Foundations of Computer Science*, 22(2):277–288, 2011.
4. M.-P. Béal and D. Perrin. A quadratic upper bound on the size of a synchronizing word in one-cluster automata. In *Developments in Language Theory*, volume 5583 of *LNCS*, pages 81–90. Springer, 2009.
5. M. Berlinkov and M. Szykuła. Algebraic Synchronization Criterion and Computing Reset Words. In *Mathematical Foundations of Computer Science*, volume 9234 of *LNCS*, pages 103–115. Springer, 2015.
6. M. V. Berlinkov. On the probability to be synchronizable. `http://arxiv.org/abs/1304.5774`, 2013.

7. M. V. Berlinkov. Synchronizing Quasi-Eulerian and Quasi-one-cluster Automata. *International Journal of Foundations of Computer Science*, 24(6):729–745, 2013.
8. M. V. Berlinkov. Approximating the Minimum Length of Synchronizing Words Is Hard. *Theory of Computing Systems*, 54(2):211–223, 2014.
9. M. V. Berlinkov. On Two Algorithmic Problems about Synchronizing Automata. In *Developments in Language Theory*, LNCS, pages 61–67. Springer, 2014.
10. M. T. Biskup. Shortest Synchronizing Strings for Huffman Codes. In *Mathematical Foundations of Computer Science*, volume 5162 of *LNCS*, pages 120–131. Springer, 2008.
11. M. T. Biskup and W. Plandowski. Shortest synchronizing strings for Huffman codes. *Theoretical Computer Science*, 410(38-40):3925–3941, 2009.
12. A. Carpi and F. D'Alessandro. Independent sets of words and the synchronization problem. *Advances in Applied Mathematics*, 50(3):339–355, 2013.
13. A. Carpi and F. D'Alessandro. Černý-like problems for finite sets of words. In *Proceedings of the 15th Italian Conference on Theoretical Computer Science, Perugia, Italy, September 17-19, 2014.*, pages 81–92, 2014.
14. J. Černý. Poznámka k homogénnym experimentom s konečnými automatmi. *Matematicko-fyzikálny Časopis Slovenskej Akadémie Vied*, 14(3):208–216, 1964. In Slovak.
15. L. Dubuc. Sur les automates circulaires et la conjecture de Černý. *Informatique théorique et applications*, 32:21–34, 1998. In French.
16. D. Eppstein. Reset sequences for monotonic automata. *SIAM Journal on Computing*, 19:500–510, 1990.
17. C. F. Freiling, D. S. Jungreis, F. Theberge, and K. Zeger. Almost all complete binary prefix codes have a self-synchronizing string. *IEEE Transactions on Information Theory*, 49(9):2219–2225, 2003.
18. P. Gawrychowski and D. Straszak. Strong inapproximability of the shortest reset word. In *Mathematical Foundations of Computer Science*, volume 9234 of *LNCS*, pages 243–255. Springer, 2015.
19. M. Gerbush and B. Heeringa. Approximating minimum reset sequences. In *Implementation and Application of Automata*, volume 6482 of *LNCS*, pages 154–162. Springer, 2011.
20. H. Jürgensen. Synchronization. *Information and Computation*, 206(9-10):1033–1044, 2008.
21. J. Kari. Synchronizing finite automata on Eulerian digraphs. *Theoretical Computer Science*, 295(1-3):223–232, 2003.
22. J. Kari and M. V. Volkov. Černý's conjecture and the road coloring problem. In *Handbook of Automata*. European Science Foundation, 2013.
23. A. Kisielewicz and M. Szykuła. Synchronizing Automata with Large Reset Lengths. `http://arxiv.org/abs/1404.3311`, 2014.
24. D. E. Knuth. *The Art of Computer Programming, Volume 4, Fascicle 4: Generating All Trees–History of Combinatorial Generation (Art of Computer Programming)*. Addison-Wesley Professional, 2006.
25. C. Nicaud. Fast synchronization of random automata. `http://arxiv.org/abs/1404.6962`, 2014.
26. J. Olschewski and M. Ummels. The complexity of finding reset words in finite automata. In *Mathematical Foundations of Computer Science*, volume 6281 of *LNCS*, pages 568–579. Springer, 2010.
27. J.-E. Pin. Utilisation de l'algèbre linéaire en théorie des automates. In *Actes du 1er Colloque AFCET-SMF de Mathématiques Appliquées II*, AFCET, pages 85–92, 1978. In French.
28. J.-E. Pin. On two combinatorial problems arising from automata theory. In *Proceedings of the International Colloquium on Graph Theory and Combinatorics*, volume 75 of *North-Holland Mathematics Studies*, pages 535–548, 1983.
29. B. Steinberg. The averaging trick and the Černý conjecture. *International Journal of Foundations of Computer Science*, 22(7):1697–1706, 2011.
30. B. Steinberg. The Černý conjecture for one-cluster automata with prime length cycle. *Theoretical Computer Science*, 412(39):5487–5491, 2011.
31. M. V. Volkov. Synchronizing automata and the Černý conjecture. In *Language and Automata Theory and Applications*, volume 5196 of *LNCS*, pages 11–27. Springer, 2008.
32. A. S. Voynov and V. Yu. Protassov. Compact noncontraction semigroups of affine operators. *Sbornik: Mathematics*, 206(7):921–940, 2015.